

UNITED STATES PATENT APPLICATION
FOR
ADAPTIVE LOAD BALANCING

INVENTORS:

CHRISTOPHER MURRAY
JOHN ZAMICK

PREPARED BY:

HICKMAN PALERMO TRUONG & BECKER LLP
1600 WILLOW STREET
SAN JOSE, CALIFORNIA 95125
(408) 414-1080

"Express Mail" mailing label number EV322192583US

Date of Deposit November 12, 2003

ADAPTIVE LOAD BALANCING

FIELD OF THE INVENTION

[0001] The present invention relates to load balanced servers in a network. The invention specifically relates to adaptive load balancing.

BACKGROUND OF THE INVENTION

[0002] In a client-server computer system, clients rely on servers to provide needed services. In the simplest form of these systems, a single server serves multiple clients. If this is the case, then any degradation in the quality of service (QoS) provided by the server, or failure of the server, will result in poor or failed service at each of its clients.

[0003] In many cases, however, this single point of failure is unacceptable. Therefore, systems are often built such that multiple servers are available to service clients, and clients are able to change (“failover”) from one server to another. For example, if a client detects that a server fails to respond, then the client can failover to another server providing the same service.

[0004] One approach for detecting the need for failover is to use a timeout mechanism configured on the client. In this timeout approach, given a particular request, the client will wait time T for a response from the server and will retry the request R times, again waiting time T for each retry. In a situation where the server cannot respond in time T to the request, either because the server is down (has failed), or the QoS has degraded, then the client waits for a total time of $R*T$ without a response to the request and then fails over to another server.

[0005] A problem with the timeout approach is that the client wastes the total time to failover of $R*T$. Another problem with the timeout approach is that failover time is constant for a particular client. In many cases, a server’s speed of response is dictated by the server’s operating conditions, including network conditions. In the timeout approach, the client’s

timeout value does not adapt and therefore the client's QoS suffers under changing conditions.

[0006] A second problem with the timeout approach is that it increases network traffic. Depending on implementation, O(R) messages per client will be passed when failover is needed.

[0007] Once a server has "timed out" a predefined number of times for a particular client, the client fails over to a second server. This second server is typically chosen from a preconfigured list of alternative servers on the client. A problem with this configured failover approach is that the choice of server to which to failover is based on a fixed list and not on network conditions or the operating conditions of the original server or the servers to which the client could failover.

[0008] Another approach is to use a load balancer to handle failover. A load balancer routes messages between clients and servers, acting as a single point of contact for multiple clients and allowing those clients to be served by multiple servers. In many cases, a client must be served by the same server for all related messages. In such cases, the load balancer must make client-server relationships "sticky" even when using a stateless protocol such as hypertext transfer protocol (HTTP) that does not inherently support maintaining long-duration connections of clients to servers. A load balancer makes a client-server session sticky by either keeping state for each client session, thereby keeping track of the routing of messages between clients and servers, or otherwise determining, for each message for each client-server relationship, to which client-server relationship that message corresponds.

[0009] A problem with the load balancer approach is that the implementations of stickiness algorithms are computationally expensive, memory intensive, and difficult to deploy. A related problem with the load balancer approach is that it requires at least one

separate process, the load balancer. If a client could failover correctly on its own, then there would be no need for a load balancer and load balanced client-server systems as a whole could be simpler.

[0010] Another problem with the load balancer approach is that determining the server to which to failover is based on a preconfigured list on the load balancer and not on network conditions or the operating conditions of the original server or the servers to which the client could failover.

[0011] From the above Background and in the upcoming description it will be clear that there is a need for a system for adaptive load balancing that overcomes the problems of clients failing over to alternative servers without considering the first servers operating conditions, including network conditions, or the other server's operating conditions; and needing a separate process for load balancing.

[0012] The approaches described in this section are approaches that could be pursued, but not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated, it should not be assumed that any of the approaches described in this section qualify as prior art merely by virtue of their inclusion in this section.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0014] FIG. 1 depicts a block diagram of example architectural components and layout of a load balanced system.

[0015] FIG. 2 depicts a flow diagram of an example method for determining when to send a behavior modification hint.

[0016] FIG. 3 depicts a flow diagram of an example method for determining appropriate reaction to a behavior modification hint.

[0017] FIG. 4 depicts a block diagram of example architectural elements of a load balanced authentication, authorization, and accounting (AAA) server that performs the foregoing steps.

[0018] FIG. 5 is a block diagram that illustrates a computer system upon which an embodiment of the invention may be implemented.

DETAILED DESCRIPTION OF THE INVENTION

[0019] A method and apparatus for adaptive load balancing is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent to one with ordinary skill in the art, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

[0020] 1.0 GENERAL OVERVIEW

[0021] 2.0 STRUCTURAL OVERVIEW

[0022] 3.0 FUNCTIONAL OVERVIEW

[0023] 3.1 OPERATING CONDITIONS

[0024] 3.2 DETERMINING WHEN TO SEND A BEHAVIOR

[0025] MODIFICATION HINT

[0026] 3.3 REACTING TO A BEHAVIOR MODIFICATION HINT

[0027] 3.4 AN EXAMPLE EMBODIMENT OF ADAPTIVE LOAD

[0028] BALANCING FOR AN AAA SERVER

[0029] 3.5 FUNCTIONAL ARCHITECTURE

[0030] 4.0 HARDWARE OVERVIEW

[0031] 5.0 EXTENSIONS AND ALTERNATIVES

1.0 GENERAL OVERVIEW

[0032] The needs identified in the foregoing Background, and other needs and objects that will become apparent for the following description, are achieved in the present invention, which comprises, in one aspect, a method for adaptive load balancing including the steps of monitoring a server's operating conditions; determining, based on the server's operating conditions, when to send a behavior modification hint to one or more clients that are being served by the server; generating the behavior modification hint based on the server's operating conditions; and sending the behavior modification hint to the one or more clients. In a related feature, the server is an AAA server and the one or more clients are AAA clients. In a related feature, the step of sending the behavior modification hint comprises sending a RADIUS message containing the behavior modification hint in a vendor specific attribute within the RADIUS message.

[0033] In a related feature, the step of sending the behavior modification hint comprises sending a particular message containing the behavior modification hint to a particular client of the one or more clients, where the particular message is a response message to a request message sent by the particular client to the server. In a related feature, the step of monitoring the server's operating conditions comprises monitoring at least one of average transaction request processing time, CPU usage percentage, memory usage percentage, network conditions, and number of processes running.

[0034] In a related feature, the method further includes the step of determining the one or more clients to which to send the message based on a predefined list of clients. In a related feature, the method further includes the step of determining the one or more clients to which to send the message based on a network device group. In a related feature, the method further

includes the step of determining the one or more clients to which to send the message based on operating conditions for the server relative to each of the one or more clients. In a related feature, the server is one of multiple servers providing a particular service; the behavior modification hint comprises a suggestion of one or more alternative servers; and the method further comprises the step of determining the one or more alternative servers based on the set of operating conditions for each server of the multiple servers. In a related feature, the step of determining the one or more alternative servers further comprises the server obtaining the operating conditions of the multiple servers over a network.

[0035] In a related feature, the step of determining when to send a behavior modification hint is based on network conditions of a network providing communication between the server and the one or more clients, where the network conditions comprise at least one of ping time from the server to a computer on the network; round trip time of a message sent to a particular client; quality of service guaranteed to one or more clients; and operating conditions of a device on the network used to route messages. In a related feature, the step of sending a behavior modification hint further comprises the steps of sending a code to the one or more clients; and generating the code based on why it was determined to send a message to the one or more clients. In a related feature, the step of determining when to send a behavior modification hint is based on a scheduled event related to the server. In a related feature, the scheduled event related to the server is selected from the group consisting of server shutdown, server maintenance, and server backup. In a related feature, the step of determining when to send a behavior modification hint is based on a server detecting that a particular client has sent one or more retry messages, where a retry message is a second or subsequent message corresponding to a particular request for service from a particular client.

[0036] In another aspect, techniques are provided for a method for adaptive load balancing including the steps of receiving a behavior modification hint from a first server providing a first service, where the behavior modification hint comprises the first server's operating conditions; and altering one or more functional aspects of a client based on the behavior modification hint, where the one or more functional aspects of the client comprise at least one of a configured timeout value for the first server for the first service and a preferred server setting for the first service. In a related feature, the step of receiving a behavior modification hint comprises receiving a particular message containing the behavior modification hint from the first server, where the particular message is sent by the first server in response to a request message sent by the client to the first server.

[0037] In a related feature, the step of altering one or more functional aspects of a client comprises altering the configured timeout value for the first server for the first service. In a related feature, the method further includes the step of generating a new timeout value based on the first server's operating conditions. In a related feature, the behavior modification hint contains a list of one or more alternative servers and the step of altering one or more functional aspects of a client comprises altering the preferred server setting for the first service based on the list of one or more alternative servers. In a related feature, a second server is one of the servers in the list of one or more alternative servers and the method further comprises the step of connecting to the second server. In a related feature, the method further includes the step of generating a new timeout value based on the second server's operating conditions.

[0038] In a related feature, the step of receiving a behavior modification hint further comprises the steps of receiving a RADIUS message containing the behavior modification

hint in a Vendor Specific Attribute (VSA) within the RADIUS message and interpreting the behavior modification hint contained within the RADIUS message.

[0039] In another aspect, a computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform any of the foregoing steps.

2.0 STRUCTURAL OVERVIEW

[0040] FIG. 1 depicts a block diagram of example architectural components and layout of a load balanced system.

[0041] One or more supplicants 101A, 101B, 101C are communicatively coupled to network devices 105A, 105B. In one embodiment, communication of supplicants 101A, 101B, 101C with network devices 105A, 105B is over a network 155. In various embodiments, the network 155 is a wireless network, dial up access, the Internet, a local area network (LAN), or any other communication network. In various embodiments, the network device 105A, 105B are wireless access points, virtual private network devices, network access servers, switches, routers, or any other appropriate devices.

[0042] The network devices 105A, 105B are communicatively coupled to a LAN 150. In various embodiments, the LAN 150 is a wireless network, dial up access, the Internet, or any other appropriate communications network. The network device 105A is also communicatively coupled to a log 135. In various embodiments, the log is a database, a flat file, or any other appropriate storage.

[0043] One or more servers 110A, 110B, 110C are communicatively coupled to the LAN 150 and to respective logs 136A, 136B, 136C. In various embodiments, the servers are AAA servers, application servers, database servers, or any other servers that can support load balancing. According to one embodiment of the techniques herein described, the servers

110A, 110B, and 110C are AAA servers and the network devices 105A, 105B are AAA clients.

[0044] Consider this example of a functioning system of FIG. 1. Network device 105A acts as an access regulator for a supplicant 101A, controlling what the supplicant 101A can reach in the rest of the system 100. The network device 105A accounts for all of the activity that passes through it via a log 135. When supplicant 101A requests a service from a server 110A in the system 100, the network device 105A communicates with the servers 110A to forward the request from supplicant 101A through the LAN 150. All activity at the server 110A is accounted for in a log 136A.

3.0 FUNCTIONAL OVERVIEW

[0045] The following functional description assumes no particular hardware, operating system, software system, or other detail of an implementation. Additionally, the flow diagrams presented are examples of possible algorithmic flow and in no way limit the scope of the invention. Embodiments of the invention can be practiced in many ways in many disparate hardware and software environments and using different algorithmic flow.

[0046] One approach herein uses a preemptive method to indicate to clients that services from the server are going to degrade or fail and that the clients should alter their expectations of that server or failover to alternative servers.

3.1 OPERATING CONDITIONS

[0047] As will be described in more detail below, in various embodiments, a server sends a behavior modification hint based on the operating conditions of the server. The operating conditions include any aspect of the server itself or its network environment that can affect the server's ability to serve a client. In one embodiment, a server's operating conditions

comprise detecting that a particular client has sent one or more retry messages, where a retry message is a second or subsequent message corresponding to a particular request for service. In various embodiments, a server's operating conditions include average transaction time for a particular type of request for the server or average transaction time for a particular type of request for one or more other servers to which the server is communicatively coupled. In various embodiments, a server's operating conditions comprise CPU (central processing unit) usage percentage, memory usage percentage, network conditions, or number of processes running. In related embodiments, network conditions are computed as the round trip time for a particular request less the transaction time for the particular request or network ping time between the client and server.

[0048] In one embodiment, a server's operating conditions is computed relative to a particular client. For example, in the context of FIG. 1, where the server 110A is an AAA server and the network device 105A is an AAA client, the AAA server 110A determines CPU usage, which is a parameter relative to all clients, and network ping time relative to network device 105A.

[0049] In various embodiments, a server's operating conditions may include the schedule for server shutdown, server maintenance, server backup or any other scheduled event related to the server. In one embodiment, a server's operating conditions include operating conditions of one or more other servers to which the server is communicatively coupled. In various related embodiments, the server obtains the operating conditions of the one or more other servers over a network, via file transfer protocol (FTP), via HTTP, secure HTTP (HTTPS), TCP/IP (Transaction Control Protocol/ Internet Protocol) sockets, or other appropriate data transport mechanisms.

[0050] Some embodiments described herein determine whether the operating conditions of a server meet certain criteria. In various embodiments, determining whether operating conditions meet certain criteria comprises detecting whether a particular client has sent one or more retry messages, where a retry message is a second or subsequent message corresponding to a particular request for service; determining whether network ping time from the server to the client is over or under certain limits; determining whether average transaction time for a particular type of request for a server is over or under certain limits; or determining whether an average transaction time for a particular type of request for servers communicatively coupled to a particular server are over or under certain limits.

[0051] In various embodiments, determining whether operating conditions meet certain criteria comprises determining whether a server's CPU usage percentage is over or under certain limits, whether the server's memory usage percentage is over or under certain limits, whether the server's network conditions are better or worse than certain predefined thresholds, or whether the number of processes running on the server is over or under certain limits. In a related embodiment, determining whether the server's network conditions are above or below certain thresholds comprises comparing a predefined threshold to either a function of the round trip time for a particular request and the reported transaction time for the particular request or the ping time between a server and a client. In various embodiments, determining whether operating conditions meet certain criteria includes determining when the server will shutdown, have maintenance, perform a backup, or perform any other scheduled event related to the server.

[0052] In one embodiment, determining whether a particular server's operating conditions meet certain criteria comprises determining whether other communicatively coupled servers' operating conditions meet certain criteria. In various related embodiments,

the server determines whether the other communicatively coupled servers' operating conditions meet certain criteria in part by obtaining the servers' operating conditions over a network, FTP, HTTP, HTTPS, TCP/IP sockets, or other appropriate data transport mechanisms.

3.2 DETERMINING WHEN TO SEND A BEHAVIOR MODIFICATION HINT

[0053] FIG. 2 depicts a flow diagram of an example method for determining when to send a behavior modification hint.

[0054] In step 210, a server's operating conditions are monitored. In one embodiment, a server monitors its own operating conditions. Alternatively, a process communicatively coupled to a server monitors the server's operating conditions.

[0055] In the context of FIG. 1, for example, a first server 110A monitors its own operating conditions and these operating conditions include one or more of CPU usage percentage, memory usage percentage, network conditions, number of processes running, and knowledge of the server's 110A maintenance cycles of server 110A. The server's operating conditions also comprise the operating conditions of servers 110B and 110C, which server 110A obtains using TCP/IP sockets.

[0056] Whether the operating conditions meet certain criteria is tested in step 220. In one embodiment, step 220 includes determining whether the operating conditions of one or more other servers meet certain criteria, where the other servers are communicatively coupled to the first server.

[0057] In various embodiments, step 220 includes determining whether CPU usage is over a certain percentage, whether memory usage is over a certain percentage, whether network ping time is higher than a predefined limit, or whether the number of processes running is over a certain limit. In various embodiments, step 220 includes determining

whether server shutdown, server maintenance, server backup or any other scheduled event related to the server could affect the ability to serve clients.

[0058] In the context of FIG. 1, for example, a server 110A determines whether its own operating conditions meet certain criteria, and the operating conditions include one or more of CPU usage percentage, memory usage percentage, network conditions, number of processes running, and knowledge of the maintenance cycles. Determining whether the server's operating conditions meet certain criteria also includes determining whether the operating conditions of servers 110B and 110C meet certain criteria. In this example, the server 110A determines whether servers 110B or 110C are in an operating condition suitable for servicing clients.

[0059] If a server's operating conditions meet certain criteria, then a behavior modification hint is sent to one or more clients at step 230. In this context, a behavior modification hint is any indication by which server suggests to clients that services from the server are going to degrade or fail and that the clients should alter their expectations of that server or failover to alternative servers. In one embodiment, the behavior modification hint is sent to a client from a server in a message that is sent in response to a request from the client to the server.

[0060] In one embodiment, in which server 110A is a AAA server, sending the behavior modification hint comprises sending a Remote Authentication Dial-In User Service (RADIUS) message containing therein a RADIUS Vendor Specific Attribute (VSA) containing the behavior modification hint. In various embodiments, the behavior modification hint is included as part of a message in Terminal Access Controller Access Control System (TACACS++) or Diameter protocols. However, the specific mechanism used to send the hint is not critical.

[0061] In various embodiments, determining the one or more clients to which to send the behavior modification hint is based on a predefined list of clients, a network device group, operating conditions for the server relative to each of the one or more clients, or on network conditions. In related embodiments, the network conditions comprise ping time from the server to a computer on the network; round trip time of a message sent to a particular client; quality of service guaranteed to one or more clients; or operating conditions of a device on the network used to route messages.

[0062] In one embodiment, the server is one of multiple servers providing a particular service; in this arrangement, the server knows the operating conditions of each of the multiple servers and a suggestion of one or more alternative servers from among the multiple servers is sent along with the behavior modification hint. In a related embodiment, the suggestion of one or more alternative servers is based on the operating conditions of each server of the multiple servers. In various embodiments, the behavior modification hint is sent to one client of the multiple clients a server serves, a proper subset of the multiple clients the server serves, or to all clients the server serves. In one embodiment, one or more reason codes are sent with the behavior modification hint. The reason codes indicate a reason why the server is providing a behavior modification hint. These reason codes are determined based on which operating conditions met which criteria. Client-side software or other mechanisms may use the reason codes to determine how to process the behavior modification hint.

[0063] In the context of FIG. 1, for example, a server 110A sends a behavior modification hint to all of its clients in order to inform them that the server's memory usage is over a certain limit, and the behavior modification hint includes a reason code

corresponding to the memory usage being over a certain limit. The behavior modification hint also includes a list of alternative servers 110B and 110C and their operating conditions.

[0064] After a behavior modification hint is sent to one or more clients in step 230, or if a server's operating conditions do not meet certain criteria in step 220, then the server's operating conditions are monitored, step 210. In one embodiment, the server's performance is continually monitored.

[0065] Various embodiments of FIG. 2 overcome the need for a client to use only a timeout mechanism for failover and allows servers or processes communicatively coupled thereto to indicate to clients or processes communicatively coupled thereto the state of operating conditions for the server, reasons for sending behavior modification hints, and a list of alternative servers to which the clients can failover and eliminate the need for a separate process to perform load balancing. These indications enable a client to make an informed decision about when and to which server to failover. Moreover, various embodiments reduce the network traffic associated with timeout, failover, and reconnection.

[0066] Whereas FIG. 2 depicts a certain flow of events, the invention is not limited to these steps or this flow. Additional steps could be performed, steps could be left out, and the steps could be performed in parallel or in a different order.

3.3 REACTING TO A BEHAVIOR MODIFICATION HINT

[0067] FIG. 3 depicts a flow diagram of an example method for determining appropriate reaction to a behavior modification hint.

[0068] A behavior modification hint is awaited in step 310. In various embodiments, a client awaits receiving a behavior modification hint as part of a message in any appropriate protocol, such as RADIUS, TACACS+, or Diameter. In various embodiments, a client awaits receipt of a behavior modification hint or a process thereto communicatively coupled

awaits arrival of a behavior modification hint at the client. In one embodiment, a client awaits a behavior modification hint after sending a request for service to a server. In a related embodiment, the client awaits the behavior modification hint at least in part by investigating the responses sent by the server. For example, in the context of FIG. 1 where the server 110A is an AAA server, a network device 105A, an AAA client, awaits receipt of a behavior modification hint by investigating the contents of the Vendor Specific Attributes in RADIUS messages sent in response to the client's request for service.

[0069] Once the behavior modification hint has arrived, it is received, step 320. In various embodiments, receiving a behavior modification hint includes receiving the behavior modification hint via a network connection from a server or a process communicatively coupled to a server or polling for the behavior modification hint at a known location and downloading the behavior modification hint over an appropriate network connection. In one embodiment, receiving the behavior modification hint includes storing the behavior modification hint in a computer readable medium communicatively coupled to a client.

[0070] In one embodiment, the behavior modification hint includes operating conditions. In related embodiments, the behavior modification hint contains information regarding whether the server's operating conditions meet certain criteria. In various embodiments, accompanying or contained within the behavior modification hint is one or more reason codes corresponding to the reason(s) the behavior modification hint was sent.

[0071] For example, in the context of FIG. 1 where the server 110A is an AAA server, a network device 105A, an AAA client, receives a behavior modification hint from a server 110A which contains information about the server's operating condition and suggests alternative servers is stored in a memory communicatively coupled to the client.

[0072] Once the behavior modification hint is received in step 320, then a check is made to determine whether the client needs to failover to another server, step 330. In various embodiments, the decision whether to failover is based on the operating conditions of the server that sent the behavior modification hint, an estimated time that the client would have to wait for service from the server; or the operating conditions of other servers which could provide the same service as the server which sent the behavior modification hint. In related embodiments, the operating conditions include any of the operating conditions described above. In a related embodiment, the estimated time that the client would have to wait for service from the server is based on a function of the operating conditions of the server. In one embodiment, the client makes the decision to failover based on an estimate of how long the server would take to service the client and how long each of one or more alternative servers would take to service the client.

[0073] For example, in the context of FIG. 1 where the server 110A is an AAA server, a network device 105A, an AAA client, determines whether it needs to failover based on the operating conditions of the server 110A. Specifically, the network device 105A bases the decision on whether the server's 110A CPU usage is over a predefined limit and whether the server's 110A network ping time is over a certain predefined limit.

[0074] If it is determined to failover to another server in step 330, then a connection is established to an alternative server in step 340. In one embodiment, step 340 comprises setting a preferred server setting on a client and the client using the preferred server setting to determine to which server to send the message. In various embodiments, step 340 comprises connecting to a server over a network or sending a message to a server. In various embodiments, a client chooses the alternative server based on the operating conditions of the

server; based on a preconfigured list to which the client has access; or based on a list of one or more suggested alternatives contained in the behavior modification hint in step 310.

[0075] For example, in the context of FIG. 1 where the server 110A is an AAA server, a network device 105A, an AAA client, which received a behavior modification hint from server 110A containing operating conditions of servers 110A, 110B, 110C in step 320, determines that it needs to failover based on the operating conditions of the server 110A. Then the network device 105A determines that it will failover to the server 110B based on the operating conditions of server 110B and server 110C.

[0076] After the client has failed over in step 340, or if no failover is needed in step 330, a decision is made whether to alter a timeout value on a client, step 350. In one embodiment, the decision whether to alter a timeout value is based on the operating conditions of the server that sent the behavior modification hint. In another embodiment, the decision whether to alter a timeout value is based on the operating conditions of the server to which the client failed over in step 340.

[0077] For example, in the context of FIG. 1 where the server 110A is an AAA server, a network device 105A, an AAA client, decides whether to alter a timeout value related to requests to a server 110A based on whether the server's 110A CPU usage is over a predefined limit and whether the network ping time is over a certain predefined limit. The network device 105A increases the timeout value in order to wait a longer, more appropriate amount of time for a response from the server, which is under a heavy processing load, or because of network latency problems as indicated by the high client-server ping time.

[0078] If there is a need to alter the timeout value, then the timeout value is altered at a client in step 360. In various embodiments, altering the timeout value at a client includes changing a value stored in a computer readable medium that specifies the timeout values for

all servers, changing a value stored in a computer readable medium that specifies the timeout value for the server that sent the behavior modification hint to the client; changing the amount of time that the client will wait for a particular response to a particular query from the server; or changing the amount of time that the client will wait for a particular type of service provided by a particular server. In one embodiment, altering the timeout value includes determining a new timeout value based on the operating conditions of the server. In related embodiments, determining the new timeout value comprises performing a functional composition on one or more aspects the server's operating condition.

[0079] For example, in the context FIG. 1 where the server 110A is an AAA server, a network device 105A, an AAA client, increases a timeout value associated with a particular service provided by a particular server 110A based on the server's 110A CPU usage being over a predefined limit and the server's 110A network ping time being over a certain predefined limit. The new timeout value is based on an estimate of how long the particular server will take to complete the particular service.

[0080] Various embodiments of FIG. 3 enable clients to react to the operating conditions of the servers which serve them. The clients, based on these operating conditions and behavior modification hints from the servers, can determine whether it is appropriate to wait longer for a response to a request sent to a server and when they should failover to an alternative server. Moreover, a client can choose the alternative server intelligently based on the operating conditions of these alternative servers, suggestions by the current server, or predefined lists. Moreover, various embodiments eliminate the need for a separate load balancing process and reduce the network traffic associated with timeout value, failover, and reconnection.

[0081] Whereas FIG. 3 depicts a certain flow of events, the invention is not limited to these steps or this flow. Additional steps could be performed, steps could be left out, and the steps could be performed in parallel or in a different order.

3.4 AN EXAMPLE EMBODIMENT OF ADAPTIVE LOAD BALANCING FOR AN AAA SERVER

[0082] An example system with load balanced authentication, authorization, and accounting (AAA) servers according to one embodiment of the techniques described herein and clients is described for purposes of illustrating a clear example, but other embodiments, some of which are described above, are possible. AAA servers provide the following services to clients in that environment:

[0083] Authentication: Validating the claimed identity of an end user or a device, such as a host, server, switch, router, etc.

[0084] Authorization: Granting access rights to a user, groups of users, system, or a process.

[0085] Accounting: Establishing who, or what, performed a certain action, such as tracking user connection and logging system users.

[0086] A network device 105A, an AAA client, sends an auth-request packet as provided in the RADIUS protocol to a load balanced AAA server 110A. Upon receipt of this request the AAA server 110A determines that the operating conditions it has been monitoring (step 210) meet certain criteria (step 220), which indicate that the server should send a behavior modification hint to the network device 105A in step 230. The AAA server 110A constructs a RADIUS auth-accept message with the behavior modification hint included in a RADIUS Vendor Specific Attribute in a key-value format that the client can parse. The behavior modification hint includes the AAA server's 110A CPU usage, which is higher than a

predefined threshold; the server 110A to network device 105A ping time, which is higher than a predefined threshold; a suggestion of an alternative AAA server 110B; and the alternative server's 110B CPU usage. This message containing a behavior modification hint is sent to the network device 105A (230).

[0087] The network device 105A, which had been awaiting the reply to a request (step 310), receives the RADIUS auth-accept message from the AAA server 110A (step 320). The network device 105A parses and interprets the behavior modification hint in the Vendor Specific Attribute of the RADIUS auth-accept message and determines, based on the AAA server's 110A CPU usage and ping time, that the network device 105A needs to failover to another server (step 330). The network device 105A chooses to failover to AAA server 110B based on the CPU usage information for AAA server 110B passed in the behavior modification hint. The network device 105A fails over to the AAA server 110B, in part, by sending the server a RADIUS auth-request message (step 340).

[0088] The network device 105A then increases the timeout value associated with the RADIUS auth-request sent to AAA server 110B message because the CPU usage of AAA server 110B being over a certain threshold (step 350) – that information having been received in the behavior modification hint in step 320. Subsequently, the network device 105A awaits a RADIUS auth-accept message and a behavior modification hint (step 310) from the AAA server 110B.

3.5 FUNCTIONAL ARCHITECTURE

[0089] FIG. 4 depicts a block diagram of example architectural elements of a load balanced AAA server that performs the foregoing steps.

[0090] In various embodiments, a server has multiple services. The administration service 410 provides a built-in web server for AAA administration of the multiple

simultaneous sessions within the server. The authorization service 420 authenticates users, grants or denies service privileges, manages AAA databases, and handles external database authentication forwarding. The database synchronization service 430 manages database synchronization and replication to other AAA servers. The logging service 440 monitors and records user and administrator activities and activities related to backups and restoration, database replication, synchronizations, TACACS+ and RADIUS communication, VOIP activities, and any other service accounting needed. The TACACS+ service 450 and RADIUS service 460 handle communication and parsing of messages passed among devices and services. The monitoring service 470, monitors status of AAA services and server resources, records and reports all critical errors to logs, sends e-mail alerts to administrators noting any potential problems, automatically detects and restarts AAA services, and scrutinizes login frequency of users.

[0091] The steps of FIG. 2 may be implemented in a behavior modification hint signaler 480. In various embodiments, the foregoing steps are performed by one or more of the services 410, 420, 430, 440, 450, 460, 470; are performed entirely by the service 480; or are performed by the service, 480, in combination with the services one or more of the services 410, 420, 430, 440, 450, 460, 470. For example, in the context of FIGs. 1 where the server 110A is an AAA server, as part of the AAA server 110A, a monitoring service 470 provides information regarding the operating conditions of the AAA server 110A to a behavior modification hint signaler 480, and when the operating conditions meet certain criteria, the behavior modification hint signaler 480 constructs a behavior modification hint to be sent by the RADIUS service 460 in a VSA to one or more network devices 105A, 105B (AAA clients) to indicate that services from the server are going to degrade or fail and that the clients should alter their expectations of that server or failover to alternative servers.

[0092] The services listed in FIG. 4 do not assume any particular hardware configuration. The services can run as part of a single thread or process, can be separate threads or processes on the same physical computer, or can be running on multiple computers.

4.0 HARDWARE OVERVIEW

[0093] FIG. 5 is a block diagram that illustrates a computer system upon which an embodiment of the invention may be implemented. Computer system 500 includes a bus 502 or other communication mechanism for communicating information, and a processor 504 coupled with bus 502 for processing information. Computer system 500 also includes a main memory 506, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 502 for storing information and instructions to be executed by processor 504. Main memory 506 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 504. Computer system 500 further includes a read only memory (ROM) 508 or other static storage device coupled to bus 502 for storing static information and instructions for processor 504. A storage device 510, such as a magnetic disk or optical disk, is provided and coupled to bus 502 for storing information and instructions.

[0094] Computer system 500 may be coupled via bus 502 to a display 512, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 514, including alphanumeric and other keys, is coupled to bus 502 for communicating information and command selections to processor 504. Another type of user input device is cursor control 516, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 504 and for controlling cursor movement on display 512. This input device typically has two degrees of freedom in two

axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0095] The invention is related to the use of computer system 500 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 500 in response to processor 504 executing one or more sequences of one or more instructions contained in main memory 506. Such instructions may be read into main memory 506 from another computer-readable medium, such as storage device 510. Execution of the sequences of instructions contained in main memory 506 causes processor 504 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

[0096] The term “computer-readable medium” as used herein refers to any medium that participates in providing instructions to processor 504 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 510. Volatile media includes dynamic memory, such as main memory 506. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 502. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0097] Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punchcards, papertape, any other physical medium with patterns of holes, a

RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0098] Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 504 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 500 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 502. Bus 502 carries the data to main memory 506, from which processor 504 retrieves and executes the instructions. The instructions received by main memory 506 may optionally be stored on storage device 510 either before or after execution by processor 504.

[0099] Computer system 500 also includes a communication interface 518 coupled to bus 502. Communication interface 518 provides a two-way data communication coupling to a network link 520 that is connected to a local network 522. For example, communication interface 518 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 518 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 518 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0100] Network link 520 typically provides data communication through one or more networks to other data devices. For example, network link 520 may provide a connection through local network 522 to a host computer 524 or to data equipment operated by an Internet Service Provider (ISP) 526. ISP 526 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the “Internet” 528. Local network 522 and Internet 528 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 520 and through communication interface 518, which carry the digital data to and from computer system 500, are exemplary forms of carrier waves transporting the information.

[0101] Computer system 500 can send messages and receive data, including program code, through the network(s), network link 520 and communication interface 518. In the Internet example, a server 530 might transmit a requested code for an application program through Internet 528, ISP 526, local network 522 and communication interface 518.

[0102] The received code may be executed by processor 504 as it is received, and/or stored in storage device 510, or other non-volatile storage for later execution. In this manner, computer system 500 may obtain application code in the form of a carrier wave.

5.0 EXTENSIONS AND ALTERNATIVES

[0103] In the foregoing specification, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is the invention, and is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any definitions expressly set forth herein for terms contained in such claims shall

govern the meaning of such terms as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.